

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.
2. Authorization for this examiner's amendment was given in a telephone interview with Borislav Nikolik on July 28, 2009.
3. The claims have been amended as follows:
 - a. Claim 2: replace claim 2 with the following:
 2. A computer program testing method for collecting internal test distribution information, and for indicating test diversity throughout source files written in the same or different programming languages; the method includes the steps:
 - parsing and instrumenting the computer program to provide an instrumented computer program;
 - executing the instrumented computer program to generate a test-distribution record and a path trace, wherein the executing step further includes the steps of:
 - dynamically updating a test-distribution record of *true/false* frequency counts associated with each conditional expression and sub-expression in the program;
 - dynamically updating a compact path trace consisting of the locations of the conditional expression in the code and their resulting Boolean values after they have been completely evaluated; and
 - possibility of altering the normal control flow of the program by discarding the resulting Boolean values of the conditional expression, dynamically generating Boolean values, substituting these generated values for the discarded ones, and continuing execution with the generated values; and
 - producing test diversity output using the test-distribution record and the path trace to indicate the internal conditional diversity, data diversity, and path diversity of the program.
 - b. Claim 4: replace claim 4 with the following:

4. A computer program testing method for collecting internal test distribution information, and for indicating test diversity throughout source files written in the same or different programming languages; the method includes the steps:
 - parsing and instrumenting the computer program to provide an instrumented computer program;
 - executing the instrumented computer program to generate a test-distribution record and a path trace;
 - producing test diversity output using the test-distribution record and the path trace to indicate the internal conditional diversity, data diversity, and path diversity of the program, wherein the producing step further includes the steps of:
 - calculating the conditional diversity for a conditional expression from a test-distribution record as a distance between the even distribution of *true* and *false* condition evaluations and the actual distribution for that expression;
 - calculating the sub-conditional diversity for a conditional expression from a test-distribution record as a distance between the even distribution of *true* and *false* sub-condition evaluations and the actual distribution for each sub condition in the expression;
 - calculating the average conditional diversity by averaging the conditional diversities for all the conditional expressions; and
 - calculating the average sub-conditional diversity by averaging the sub-conditional diversities for all the conditional expressions.
- c. Claim 6: line 1, delete “[4]” and insert -- 4 --.
- d. Claim 9: replace claim 9 with the following:
 9. A computer program testing method for collecting internal test distribution information, and for indicating test diversity throughout source files written in the same or different programming languages; the method includes the steps:
 - parsing and instrumenting the computer program to provide an instrumented computer program, wherein the instrumenting step is performed by inserting a function call around the conditional expression and conditional sub-expressions in a conditional statement to:
 - dynamically evaluate conditional expressions and sub-expressions and immediately update the *true/false* counts based on the evaluation;
 - dynamically produce a compact path trace of conditional expression locations and their values; and
 - dynamically generate Boolean values, evaluate conditional expressions, discard the resulting Boolean value from the evaluation, and substitute the generated value for the discarded one;

executing the instrumented computer program to generate a test-distribution record and a path trace; and
producing test diversity output using the test-distribution record and the path trace to indicate the internal conditional diversity, data diversity, and path diversity of the program.

Reasons for Allowance

4. The following is an examiner's statement of reasons for allowance: the prior art of record fails to teach or suggest the claimed invention. Specifically, the prior art of record, Plum fails to teach or suggest an executing step further includes the steps of: dynamically updating a test-distribution record of *true/false* frequency counts associated with each conditional expression and sub-expression in the program; dynamically updating a compact path trace consisting of the locations of the conditional expression in the code and their resulting Boolean values after they have been completely evaluated; and possibility of altering the normal control flow of the program by discarding the resulting Boolean values of the conditional expression, dynamically generating Boolean values, substituting these generated values for the discarded ones, and continuing execution with the generated values as recited in claim 2. Plum fails to teach or suggest a producing step further includes the steps of: calculating the conditional diversity for a conditional expression from a test-distribution record as a distance between the even distribution of *true* and *false* condition evaluations and the actual distribution for that expression; calculating the sub-conditional diversity for a conditional expression from a test-distribution record as a distance between the even distribution of *true* and *false* sub-condition evaluations and the actual distribution for each sub condition in the expression; calculating the average conditional diversity by averaging the conditional diversities for all the conditional expressions; and calculating the average sub-conditional diversity by averaging the sub-conditional diversities for

all the conditional expressions as recited in independent claim 4. Plum fails to teach or suggest an instrumenting step performed by inserting a function call around the conditional expression and conditional sub-expressions in a conditional statement to: dynamically evaluate conditional expressions and sub-expressions and immediately update the *true/false* counts based on the evaluation; dynamically produce a compact path trace of conditional expression locations and their values; and dynamically generate Boolean values, evaluate conditional expressions, discard the resulting Boolean value from the evaluation, and substitute the generated value for the discarded one as recited in independent claim 9. Plum fails to teach or suggest a method for software testing including the steps of: reporting conditional diversities computed using the counts from a data structure indicating the number of times conditional expressions and sub-expressions in conditional statements evaluate to true/false as a distance between the even distribution and the actual distribution of counts; and reporting data diversity as an average of individual diversities, calculated as a percentage of test suites that have distinct conditional diversities for a conditional expression as recited in independent claim 12.

5. Any comment considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

/Jue S Wang/
Examiner, Art Unit 2193